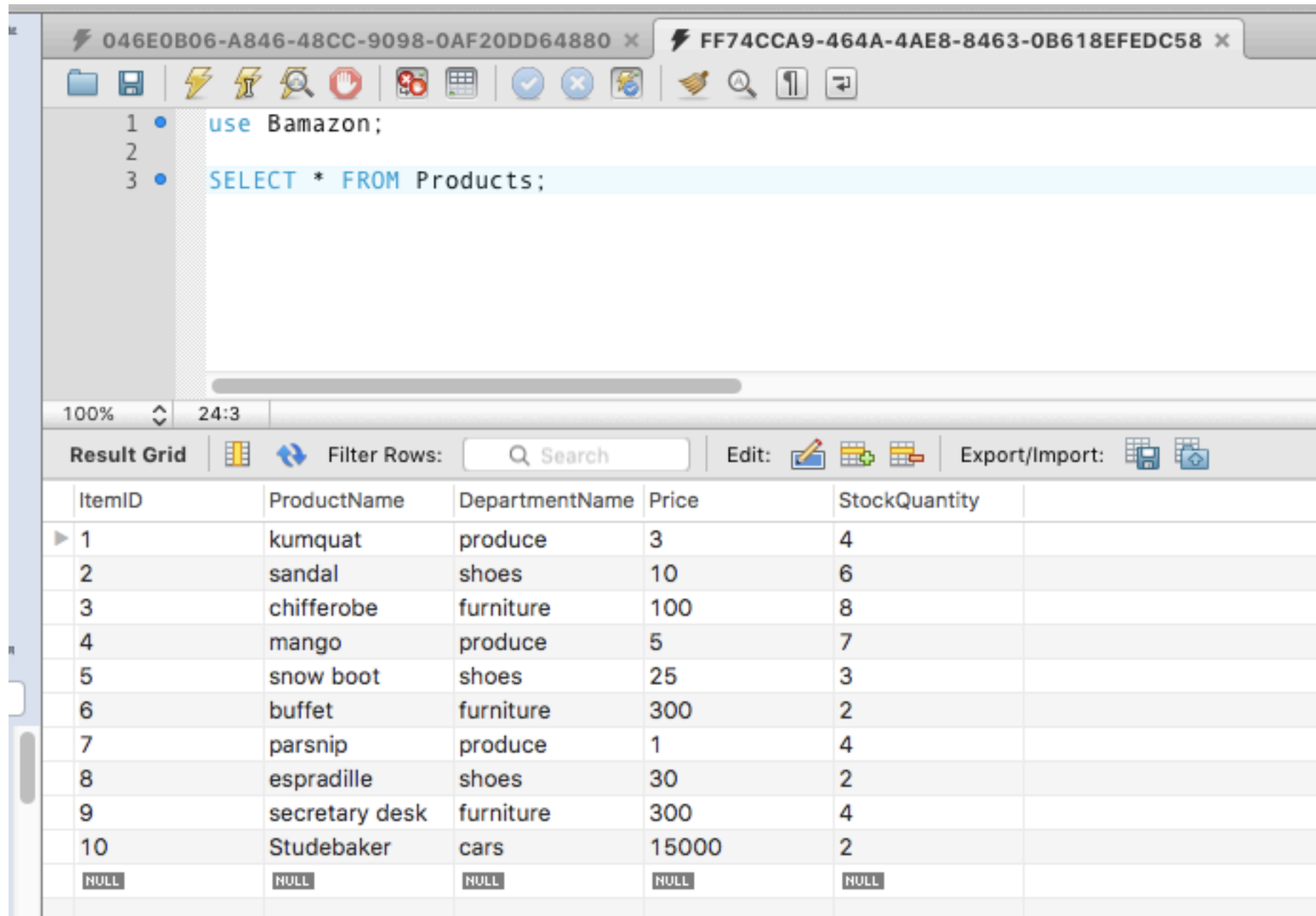


# Week12 - Bamazon

Screen shots  
Jonathan Jackson

## Products Table in Bamazon database. Note sandals quantity is 6.



The screenshot shows a database client window with two tabs. The active tab contains a SQL query: `use Bamazon;` followed by `SELECT * FROM Products;`. Below the query editor, the 'Result Grid' tab is selected, displaying a table with 6 columns: ItemID, ProductName, DepartmentName, Price, and StockQuantity. The table contains 10 rows of data, with the second row (ItemID 2) representing 'sandal' with a quantity of 6. The interface includes a toolbar with various icons for file operations, a search bar, and an 'Export/Import' section.

ItemID	ProductName	DepartmentName	Price	StockQuantity
1	kumquat	produce	3	4
2	sandal	shoes	10	6
3	chifferobe	furniture	100	8
4	mango	produce	5	7
5	snow boot	shoes	25	3
6	buffet	furniture	300	2
7	parsnip	produce	1	4
8	espradille	shoes	30	2
9	secretary desk	furniture	300	4
10	Studebaker	cars	15000	2
NULL	NULL	NULL	NULL	NULL

**Start Bamazon Customer node app. The app loads the current inventory from the Products table.**

```
[Jonathans-MacBook-Air:Week12-Bamazon Jonathan$ node BamazonCustomer.js  
-----  
Welcome to Bamazon! Please peruse our inventory.  
-----  
1 | kumquat | produce | $3 | Quantity: 4  
-----  
2 | sandal | shoes | $10 | Quantity: 6  
-----  
3 | chifferobe | furniture | $100 | Quantity: 8  
-----  
4 | mango | produce | $5 | Quantity: 7  
-----  
5 | snow boot | shoes | $25 | Quantity: 3  
-----  
6 | buffet | furniture | $300 | Quantity: 2  
-----  
7 | parsnip | produce | $1 | Quantity: 4  
-----  
8 | espradille | shoes | $30 | Quantity: 2  
-----  
9 | secretary desk | furniture | $300 | Quantity: 4  
-----  
10 | Studebaker | cars | $15000 | Quantity: 2  
-----
```

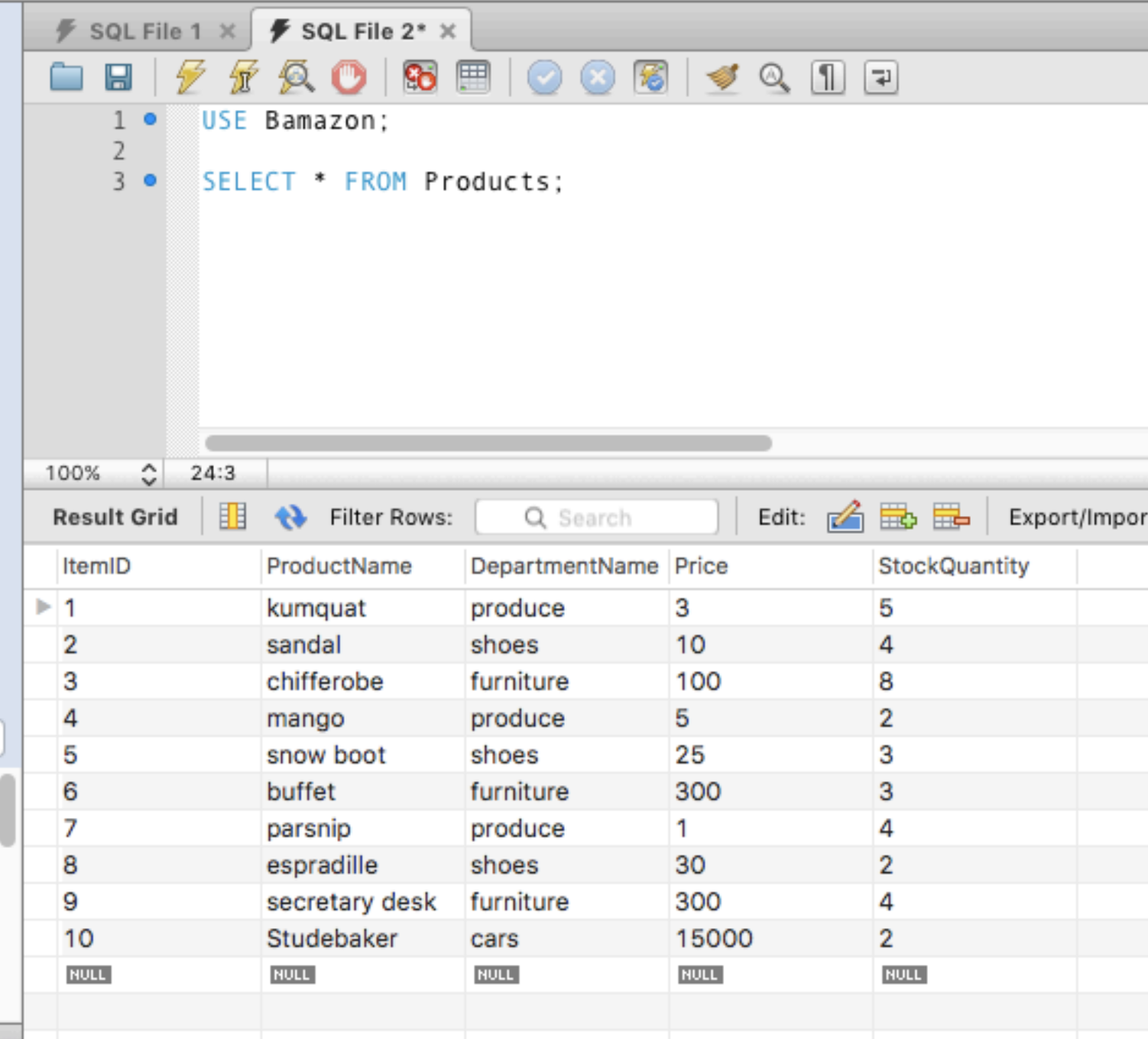
**I chose item #4 - mango. I selected 9 as the quantity. Too high. The app prompt starts over.**

```
-----  
Welcome to Bamazon! Please peruse our inventory.  
-----  
1 | kumquat | produce | $3 | Quantity: 4  
-----  
2 | sandal | shoes | $10 | Quantity: 6  
-----  
3 | chifferobe | furniture | $100 | Quantity: 8  
-----  
4 | mango | produce | $5 | Quantity: 7  
-----  
5 | snow boot | shoes | $25 | Quantity: 3  
-----  
6 | buffet | furniture | $300 | Quantity: 2  
-----  
7 | parsnip | produce | $1 | Quantity: 4  
-----  
8 | espradille | shoes | $30 | Quantity: 2  
-----  
9 | secretary desk | furniture | $300 | Quantity: 4  
-----  
10 | Studebaker | cars | $15000 | Quantity: 2  
-----  
[? Please type the ID number of the item you would like to purchase. 4  
[? How many would you like to buy? 9  
We only have 7 of that item. Please try again.  
[? Please type the ID number of the item you would like to purchase. █
```

**I chose item #2 - sandals and quantity 2. The sandals price is \$10 for a total of \$20. The app prompt then starts over.**

```
[? Please type the ID number of the item you would like to purchase. 2
[? How many would you like to buy? 2
n -----
Your total is $20. Thank you for shopping with Bamazon.
? Please type the ID number of the item you would like to purchase. █
```

**Sandal StockQuantity in the Bamazon database has updated to 4.**



The screenshot shows a SQL client window with two tabs: 'SQL File 1' and 'SQL File 2\*'. The active tab 'SQL File 2\*' contains the following SQL code:

```
1 • USE Bamazon;  
2  
3 • SELECT * FROM Products;
```

Below the code editor, the 'Result Grid' is displayed, showing the results of the query. The grid has columns for ItemID, ProductName, DepartmentName, Price, and StockQuantity. The results are as follows:

ItemID	ProductName	DepartmentName	Price	StockQuantity
1	kumquat	produce	3	5
2	sandal	shoes	10	4
3	chiffrobe	furniture	100	8
4	mango	produce	5	2
5	snow boot	shoes	25	3
6	buffet	furniture	300	3
7	parsnip	produce	1	4
8	espradille	shoes	30	2
9	secretary desk	furniture	300	4
10	Studebaker	cars	15000	2
NULL	NULL	NULL	NULL	NULL